## REMARKS

I. Overview

Claims 34-35, 37-41, 43-44, 46-47, and 49-53 are pending. Applicant has amended claim 34 to correct the typographical error identified by the Examiner.

The Examiner has rejected claims 34-35, 37-39, 41, 44, 46-47, 49-51, and 53 under 35 U.S.C. § 103(a) as being unpatentable over Polish and Gordon; and claims 40, 43, and 52 under 35 U.S.C. § 103(a) as being unpatentable over Polish, Gordon[1], and Hejna. Applicant respectfully disagrees.

II. Prior Art

There are two traditional methods of satisfying a user request to play video streaming over a network at a faster rate (e.g., fast forwarding at 2X the original speed). One example of each is described by the references relied upon by the Examiner, and each has its own drawbacks that Applicant's technology is designed to overcome.

The first method, described by Polish, uses a dumb server and places most of the burden on the client for producing the increase in video playback speed. The server in Polish simply monitors the buffer of the client, and does not know the actual speed at which the client is playing back the video. If the client's buffer is getting empty too fast, then the server sends frames to the client faster, in "bursts." If the client's buffer is too full, then the server slows down the rate of sending video to the client. The server in Polish always sends the client all of the frames of a video and the client is responsible for simulating whatever playback speed a client selects. Polish, col. 6:37-60. When a user requests a faster playback speed, the client in Polish consumes the packets from the buffer faster, such that the video plays faster. The server in Polish does not perform any

[1] Although the Office Action indicates that the Examiner is rejecting these claims under Polish and Hejna, Applicant herein assumes that this was a typographical error and that these claims, like the independent claims upon which they depend, are also rejected in view of Gordon.

type of modification of the video. The major drawback of this method is that when the user initially requests a faster playback speed, the client may consume all of the video in the buffer. This leads to a pause or hiccup in the video displayed to the user while the client waits for the server to notice that the buffer is low and refill it.

The second method, described by Gordon, uses a dumb client and places most of the burden on the server. The client in Gordon simply tells the server what the user has requested (e.g., the user pressed fast forward, the user pressed play, etc), but does not take any action to fulfill the request itself. The server in Gordon maintains multiple separate streams for each video that a user can play: one for normal playback, one for rewind, and one for fast forward. When a user requests faster playback of a video, the server switches to sending the fast forward stream. When the user later requests returning to normal playback speed, the server switches to the normal playback stream. The major drawback of this method is that the user may see an uncomfortable delay when requesting a change in the playback speed due to the latency between the client and server. After the user presses the fast forward button, the user must wait for the client to send a packet to the server indicating that the user wants faster playback, then for the server to switch to the fast forward stream, and then for the first packets of the faster playback stream to arrive from the server. This sluggish experience may make the user frustrated with the system.

## III. Applicant's Disclosure

In contrast, Applicant's disclosure describes a coordinated technique between a server and a client to increase the rate of playback of a video stream in a way that helps avoid both gaps in rendering (e.g., due to buffer underflow) and delays in responding to a user request to increase playback speed (e.g., due to network latency). (Specification, 24:8:17.) Rather than simply depending on the client or the server to accomplish the increase in playback speed, Applicant's technique has unique roles for both the client and the server to produce playback that is smooth and responds quickly to user requests.

When a user at the client requests an increase in the speed of the playback, the client can immediately start simulating the faster playback speed by skipping frames currently in its buffer that were sent for the slower playback speed. (Specification, 25:4-6.) The client also notifies the server of the new playback speed. Because the client is skipping frames, the buffer will hold less playing time of the video than desired and will more likely lead to a gap in rendering. For example, if the buffer can store 5 seconds of playback time at the slower playback speed, it would only store 2.5 seconds of playback time at double the playback speed. Because the client responds to the user's request without waiting for a response from the server, the user does not experience a delay.

When the server receives the notification of the greater playback speed, it initially sends the frames of the video for a playback speed that is greater than the new playback speed. (Specification, 25:16-22.) For example, a version of the video for a 3X playback speed will contain every third frame of the video. Thus, the sending of frames for a greater playback speed than the requested new playback speed will result in refilling the buffer of the receiving client by adding the frames at a rate faster than they need to be consumed to satisfy the requested new playback speed. In other words, by sending the video in lower resolution (e.g., 3X to be displayed at 2X), the buffer can be aggressively refilled. Because the server aggressively refills the client buffer, the client does not experience gaps in the playback because of the faster playback speed.

## IV. Rejections under 35 U.S.C. § 103(a)

Each of Applicant's claims recites a server that, following a request for a faster playback speed, initially sends data to the client at a faster rate than that requested. For example, claim 34 recites "a component of the server that, upon receiving notification of the second playback speed, initially sends to the client the stream of data that is timeline-modified for a third playback speed that is greater than the second playback speed." Claim 41 recites "a component that initially sends to the client the stream of data that is timeline-modified for a third playback speed that is greater than the second playback

speed." Claim 46 recites "after switching the rendering, initially receiving from the server the stream of data that is timeline-modified for a third playback speed that is greater than the second playback speed." Claim 53 recites "a component that upon receiving the indication, initially sends frames of the video that are timeline-modified for a third playback speed that is greater than the second playback speed."

Polish describes a computations engine that may send data to a client faster based on an expected client consumption rate. However, Polish neither teaches timeline-modification of the video nor sending data to the client at a rate faster than the expected consumption rate. In contrast, Applicant's technology does not send video or faster or slower, but rather modifies the timeline of the frames of video that are being sent, such as by choosing not to send every other frame (e.g., for a 2X video playback rate). Moreover, Applicant's technology initially sends video at a rate faster than the client has requested to prevent the client from experiencing a momentary emptying of the client's buffer that leads to a gap in playback. Nowhere does Polish describe this type of smart server behavior.

Similarly, although Gordon creates timeline-modified video streams, Gordon does not describe multiple faster video streams or initially selecting a stream that is faster than the stream requested by the client to prevent gaps in playback. Rather, Gordon only describes a single stream used for fast forward and single stream used for playback that are switched between to give the user the effect of faster video.

Therefore, the combination of Polish and Gordon fails to teach each of the elements of Applicant's claims. Accordingly, Applicant respectfully requests that this rejection be withdrawn.

In addition, the Examiner has not provided a sufficient motivation to combine Polish and Gordon to produce a system such as that of Applicant. Because Gordon uses timeline-modified frames that are always sent at the same rate regardless of the playback rate chosen by the user, Gordon has nothing to benefit from the faster sending of packets

taught by Polish. Similarly, the client described by Polish would not correctly handle the timeline-modified packets of Gordon and would attempt to play the packets faster, even though it was receiving an already sped up stream. Therefore, Applicant's claims are patentable over Polish and Gordon for this additional reason. Accordingly, Applicant respectfully requests that this rejection be withdrawn.

## V. Conclusion

Based on these remarks, Applicant respectfully requests reconsideration of this application and its early allowance. If the Examiner has any questions or believes a telephone conference would expedite prosecution of this application, the Examiner is encouraged to call the undersigned at (206) 359-3265. Applicant believes all required fees are being paid in connection with this response. However, if an additional fee is due, please charge our Deposit Account No. 50-0665, under Order No. 418268858US from which the undersigned is authorized to draw.

Dated: 12/19/2006

Respectfully submitted,

By _____

J. Mason Boswell
Registration No.: 58,388
PERKINS COIE LLP
P.O. Box 1247
Seattle, Washington 98111-1247
(206) 359-8000
(206) 359-7198 (Fax)
Attorney for Applicant